



# *CE Workgroup*

## **Embedded Linux Community Update**

**May 2021**

Tim Bird

Principal Software Engineer, Sony Electronics



CE Workgroup

# Nature of this talk...

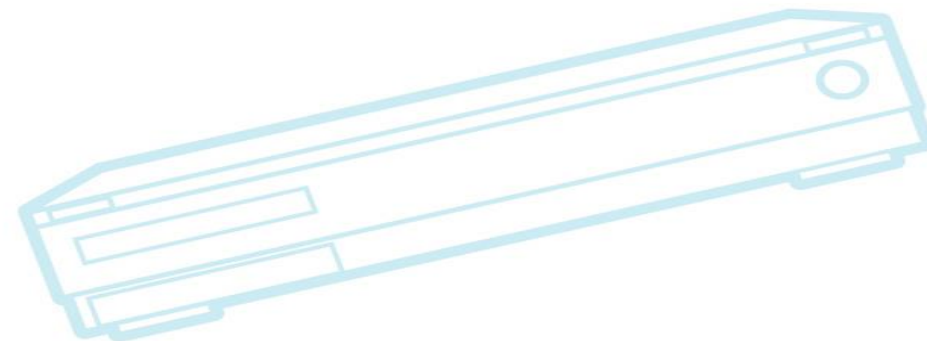
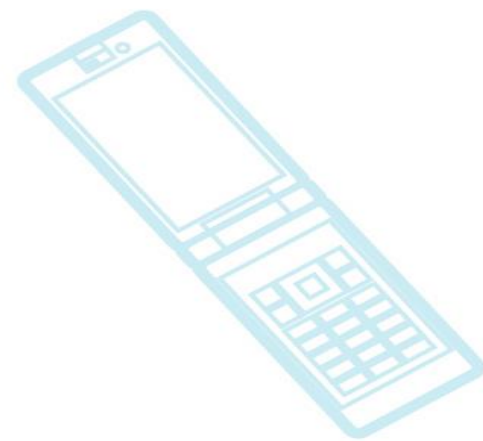
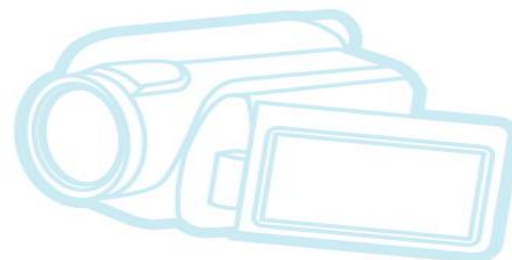
- Quick overview of lots of embedded topics
- A springboard for further research
  - If you see something interesting, you have a link or something to search for
- Some overlap with material given previously
  - I may go quickly over some older slides
- Not comprehensive!
  - Just stuff that I saw



CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Conferences  
Industry News  
Resources





CE Workgroup

# Outline

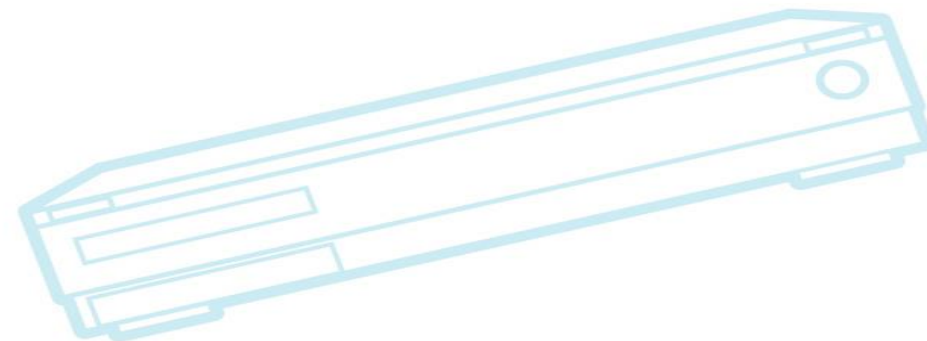
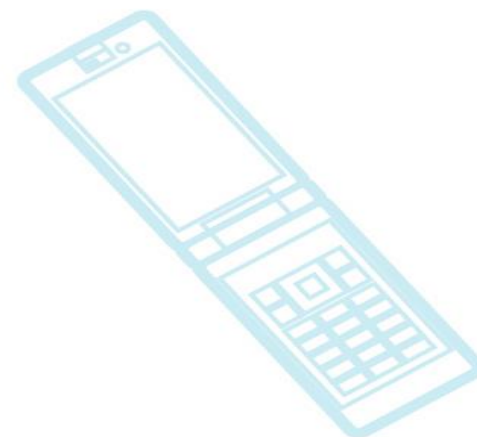
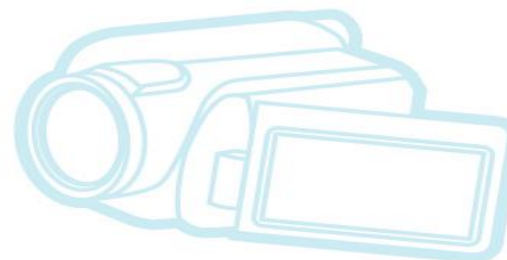
## Linux Kernel

## Technology Areas

## Conferences

## Industry News

## Resources







CE Workgroup

# Kernel Versions

- Linux v5.8 – 2 Aug 2020 – 63 days
- Linux v5.9 – 11 Oct 2020 – 70 days
- Linux v5.10 – 13 Dec 2020 – 63 days
- Linux v5.11 – 14 Feb 2021 – 63 days
- Linux v5.12 – 25 Apr 2021 – 70 days
  - v5.12 merge window delayed due to power outages from a winter storm in February.
- Linux v5.13-rc3 = current kernel this week
  - Expect 5.13 on June 27 or July 4



CE Workgroup

# Linux v5.8 (Aug 2020)

- Inline encryption for filesystems (more later)
- kgdb can now work with the boot console
  - Allows debugging earlier in the boot process
- A new generic kernel event notification system was added



CE Workgroup

# Linux v5.9 (Oct 2020)

- new 'debugfs=' command line option allows disabling debugfs (using 'off')
  - To avoid disclosing sensitive data
  - Or, it can be left enabled but not mountable with ('no-mount')
    - Data can still be read with a debugger
- Removed uninitialized\_var() macro
- initrd is deprecated (in favor of initramfs, which everyone calls initrd)
- close\_range() — new syscall to close a group of file descriptors



CE Workgroup

# Linux v5.10 (Dec 2020)

- static calls patches finally merged
  - Allows for indirect call that can be updated at runtime (via `static_call_update()`)
    - Good for tracing, and maybe other stuff
  - See <https://lwn.net/Articles/774743/#static>
- printk has a new lockless ring buffer
  - Part of a long-term overhaul of printk
  - See <https://lwn.net/Articles/800946/>
- ext4 filesystem has a new “fast commits” mode





CE Workgroup

# Linux v5.11 (Feb 2021)

- New system-call interception mechanism based on prctl()
  - Used for emulating Windows system calls
  - See <https://lwn.net/Articles/826313/>
- `epoll_pwait2()` – new syscall that supports nanosecond timeouts
- Ability to disable process migration between CPUs



# Linux v5.12 (Apr 2021)

- Support for oprofile removed
  - superceded by perf events
- “PREEMPT\_DYNAMIC” allows selecting preemption mode at boot or run time
- Dynamic thermal power management
  - Allow power usage of groups of devices to be capped to meet thermal constraints
- **Nintendo 64 support (finally!)**
  - Not sure how useful this is when the console only supports 8MB RAM, but hey!



CE Workgroup

# Linux v5.12 (cont.)

- Build system can use Clang's link-time optimization (LTO) features on ARM64 and x86 architectures
- kfence memory debugging tool has been added
- Some new perf-events features:
  - Can report on instruction latency
  - Daemon mode for long-running sessions
  - See <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=3a36281a1719>
- Support for Playstation DualSense game controllers (*yeah Sony!*)



CE Workgroup

# Linux v5.13

- Support for Control-Flow Integrity (CFI)
- Software-interrupt processing code from the Preempt RT tree was mainlined
- `logbuf_lock` used by `printk` has been removed
  - working towards lockless `printk`
- Support for Generic USB Display (GUD) driver
- BPF programs can now call (some) kernel functions directly
- `/dev/kmem` was removed (!!)
- Added new WWAN networking framework
- Landlock security module

(Many of these explained later)

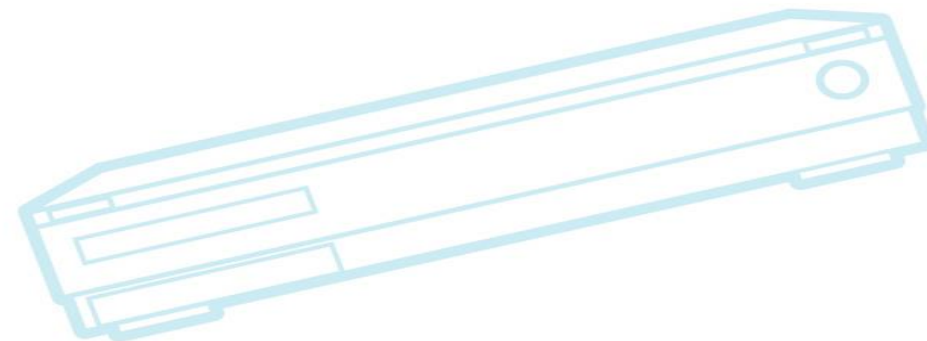
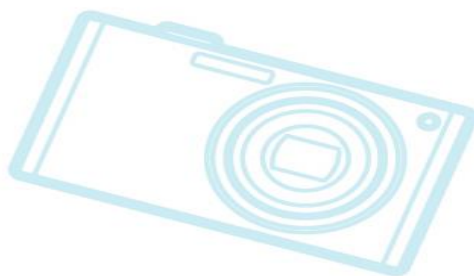
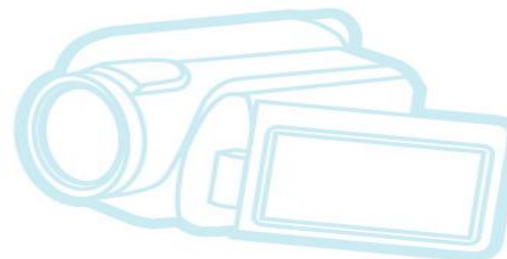
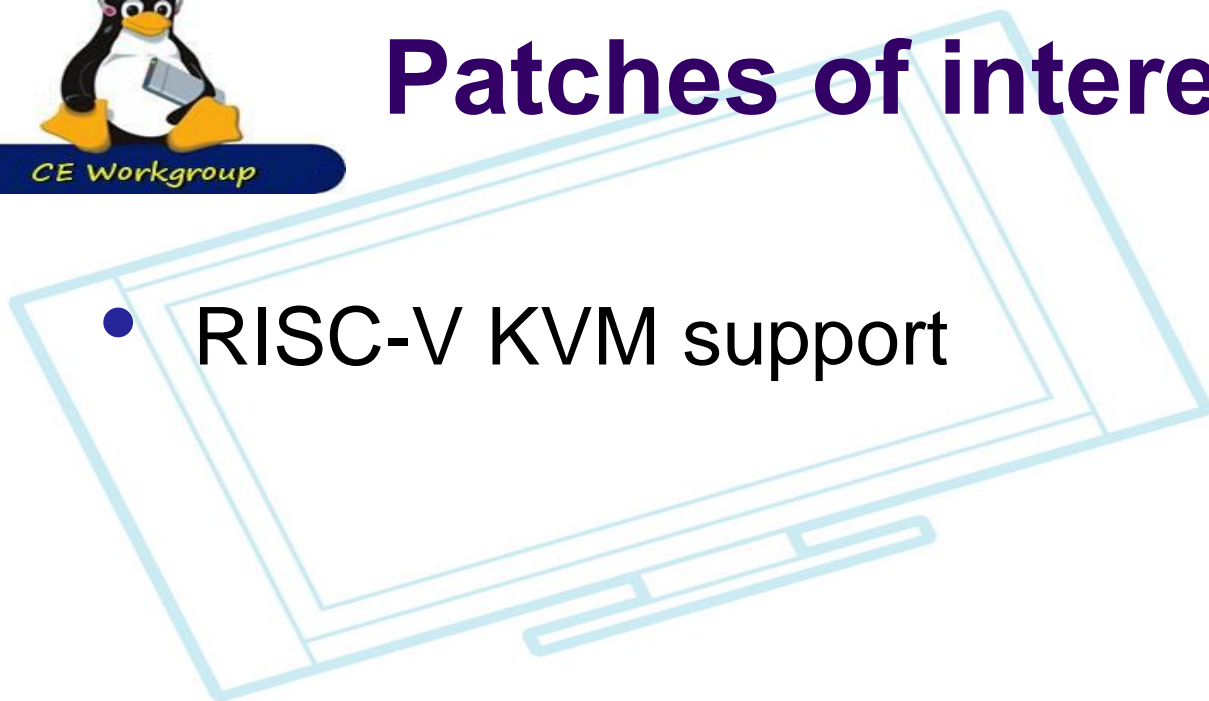




CE Workgroup

# Patches of interest (that didn't make it in)

- RISC-V KVM support





# RISC-V and KVM support

- Developer contributed patches adding KVM support for RISC-V
  - To staging directory ??
    - RISC-V maintainer won't take patches for CPU features that aren't frozen
  - “H-extension” spec approval has been delayed
    - But some vendors are shipping the instructions in hardware already
  - This is the price you pay for a mutable architecture
- Greg KH said the kernel should support shipping hardware, independent of specs
  - May take the code in staging, or have maintainer change acceptance policy (still deciding – patches deferred for now)
- See <https://lwn.net/Articles/856685/>



# Linux 5.12 developer stats

- Most active 5.12 developers
- By changesets:

Person	Changesets	Percent
Lee Jones	256	2.0%
Chris Wilson	167	1.3%
Pavel Begnkox	158	1.2%
Vladimir Oltean	120	0.9%
Christoph Hellwig	117	0.9%

- Lee Jones was also the most active changeset contributor for 5.11
  - He's been working on fixing compiler and docs-build warnings



# Linux 5.12 developer stats

- Most active 5.12 developers
- By lines of code:

Person	Lines changed	Percent
Arnd Bergmann	65277	9.4%
Po-Hao Huang	21723	3.1%
Viresh Kumar	16782	2.4%
Maximilian Luz	14520	2.1%
Andy Shevchenko	13160	1.9%





# Bug reporter stats

- Most active credited reports of bugs fixed in 5.12:

Reporter	Number of bugs	Percent
kernel test robot	184	16.1%
Syzbot	111	9.7%
Abaci Robot	107	9.4%
Dan Carpenter	44	3.9%
Hulk Robot	41	3.6%

- 4 of the top 5 were automated testing systems



# University of Minnesota ban

- U of MN banned from kernel contributions by Greg Kroah-Hartman
  - U of MN had conducted a study to see if it was possible to get patches containing security vulnerabilities into the Linux kernel, and published a research paper on their findings.
  - They claimed that no security flaws were intentionally introduced into the kernel as part of their efforts
- Kernel developers were not happy
  - Started complete re-review of 435 patches from U of MN



# How NOT to do research on an open source community...

Greg Kroah-Hartman

David A. Wheeler

May 2021



## Timeline (1 of 4)

- 2020 Aug 9..21: “Hypocrite Commits” patches sent from UMN researchers
  - Attempted to introduce vulnerabilities to see if they would be detected
  - Sent to kernel developers from false identities; without consent, notice, or ethics review
- 2020 Nov: Draft “Hypocrite Commits” paper is published
- 2020 Nov 22: Sarah Jamie Lewis calls attention to paper’s questionable ethics
- 2020 Dec 1: Lewis & others send letter to IEEE S&P, questioning ethics
- 2020 Dec [day unknown]: UMN IRB appears to give after-the-fact exemption to research on the basis that it believes the research is not human research
- 2020 Dec 15: UMN issues clarification
- 2021 Apr 6: Poor quality patches sent by UMN after ~7 months of silence
  - Raises spectre of ***continued*** attacks
- 2021 Apr 20: Greg K-H asks submitters to stop sending poor quality patches under the guise of “research on maintainers”
  - Researcher claimed new set of patches was not part of previous research
  - Greg replies, ***umn.edu submissions*** should be ***rejected*** until all figured out



## Timeline (2 of 4)

- 2021 Apr 21: Greg K-H requests review @umn.edu reverts, TAB begins review
- 2021 Apr 23: Linux Foundation sends letter to UMN requesting:
  - Id all proposals of known-vulnerable code from any U of MN experiment
  - Withdraw, from formal publication, research where subjects didn't give prior consent
  - Ensure all future U of MN experiments on people *first* have review and approval
  - Ensure all future reviews of proposed experiments on people will normally ensure the consent of those being experimented on
- 2021 Apr 24: UMN publishes "An open letter to the Linux community"
- 2021 Apr 26: UMN researchers retract "Hypocrite Commits" paper from formal publication
- 2021 Apr 27: UMN published [details on commits](#) & [replies to LF](#)
  - Paper withdrawn. UMN believes it's not "human subjects research"
  - Will do faculty ethics training in 2021-2022, explore added processes, to prevent similar situations

## Timeline (3 of 4)

- 2021 May 3: Greg K-H posts a final set of reverts, along with correct fixes
- 2021 May 5: Linux TAB [publishes detailed report](#), with due diligence audit results
  - 435 UMN commits were re-reviewed, thanks to 85 Linux kernel developers
  - Confirmed that all intentionally-vulnerable patches with vulnerabilities were rejected
    - One (“patch 1”) was intended to be vulnerable, but due to lack of understanding by the submitter, it was valid & was accepted
    - Yes, you read that correctly, you can’t make this stuff up :-)
    - Patch 1 was asked to be removed because submission was made under a false name (there have been exceptions, but true identities are still known to a subset)
  - Huge majority of the reviewed commits (349) were found to be correct
  - UMN overall patch quality relatively poor; 25 were fixed by later commits, 39 needed fixing

## Timeline (4 of 4)

- 2021 May 6: UMN meets with Greg, Kees and LF to discuss productive ways to move forward and improve
- 2021 May 6: IEEE publishes [statement](#) about how the paper violated ethical guidelines and what would be put into place to prevent it happening again
- 2021 May 7: UMN [responds to TAB report](#), verifying it is correct
  - Identifies one further set of patches from their team, using a private email address in February 2021. All were rejected by the community as they were invalid changes.
  - Stated that they had only done this for the Linux kernel, not for any other open source project:

Furthermore, we want to state unequivocally that no other Linux components or any other open software systems were affected by the 'hypocrite commits' case study or by any of our other research projects. Our "hypocrite commit" work was limited to the Linux Kernel only and consisted of only the four patches (one is valid) submitted between August 9, 2020 and August 21, 2020

- Submitting patches using false identities intending to deceive a community
- Submitting patches with known vulnerabilities (versus innocently submitting poor quality code)
- Researching on a community without notice or consent
- Every security conscious community scrambled to identify “Did UMN contribute known-vulnerable code to our project?”
  - UMN has assured us that only the Linux kernel was targeted
- Pattern of poor quality proposals (even when not intentional)
  - Asked UMN to designate a set of experienced developers to review and provide feedback on proposed kernel changes before those changes are submitted publicly; UMN agreed
  - Identical to what has been put into place for other companies



## The BAD News (1 of 2)

- It's unclear other communities without Linux kernel-level review practices would have caught these issues
- Researchers created massive amount of extra work for developer community
- IRB & other ethics process scope/definitions may not clearly cover research on community processes, even if humans are involved in those processes
  - UMN says this was a mistake & apologizes, but claims it's not "Human Subjects Research" per US federal regulations (e.g., 45 CFR 46.102)
  - Yet US [Belmont Report](#), [Common Rule](#), & [Menlo Report](#) emphasize *consent*
  - [Human Subject Regulations Decision Charts](#) suggest to us it was (for US)
  - [IEEE responded](#), "paper does not follow [ethical] guidelines"
  - [James Davis argues](#) researchers don't grok sociotechnical systems
  - NSF has been notified, need to watch this carefully

## The BAD News (2 of 2)

- Researchers sometimes do not interact with production development environments appropriately
  - Due to incentive misalignment & lack of guidance for researchers
  - TAB is working to develop guidance specifically for researchers
- Issues apply far more broadly than UMN, or US, or the Linux kernel
  - UMN promises to add ethics training & code review
  - How can we scale beyond UMN to all research?
  - How can we ensure OSS community issues are *included* in ethics decisions?

- The Linux kernel code review process **worked**
  - **All** UMN intentionally-vulnerable buggy patches were **caught** and not accepted
    - Note: One patch was accepted because it was unintentionally correct
- The Linux kernel developers rapidly reviewed all UMN contributions
  - Double-check of code should increase confidence by users & potential users
- Strong public support for the Linux kernel developers response and position, including from researchers who have been working with the kernel community for decades
- UMN apologized & actively working to prevent recurrence
- Many organizations around the world have seen the fallout from submitting intentionally weak patches and are on notice they might be banned by the community

## Where do we go from here?

- The LF and U of MN had productive discussions, including a call with key deans and leadership
- The TAB will facilitate identifying a technical mentor for U of MN similar to what we do for member companies in need of help
- U of MN will be reviewing and revising its ethical research policy, outside of IRB purview
  - U of MN promised to give the LF a heads up on a future draft
- Greg K-H and the TAB will be working with research institutions that have worked well with the kernel community to publish best practices for community research that can be a future guide for U of MN and others



# Contact Us



Learn more at [linuxfoundation.org](https://linuxfoundation.org).

For general inquiries, questions related to membership, or about our events or training offerings, please visit

[linuxfoundation.org/about/contact/](https://linuxfoundation.org/about/contact/)

## Mailing Address

548 Market St

PMB 57274

San Francisco, California

94104-5401 US

Phone/Fax: +1 415 723 9709

# Legal Notice



The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at <https://www.linuxfoundation.org/trademark-usage>, as may be modified from time to time.

Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at <https://lmi.linuxfoundation.org> for details regarding use of this trademark.

Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.

TWITTER, TWEET, RETWEET and the Twitter logo are trademarks of Twitter, Inc. or its affiliates.

Facebook and the "f" logo are trademarks of Facebook or its affiliates.

LinkedIn, the LinkedIn logo, the IN logo and InMail are registered trademarks or trademarks of LinkedIn Corporation and its affiliates in the United States and/or other countries.

YouTube and the YouTube icon are trademarks of YouTube or its affiliates.

All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.

The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at <https://www.linuxfoundation.org/privacy> and its Antitrust Policy at <https://www.linuxfoundation.org/antitrust-policy>, each as may be modified from time to time. More information about The Linux Foundation's policies is available at <https://www.linuxfoundation.org>.

Please email [legal@linuxfoundation.org](mailto:legal@linuxfoundation.org) with any questions about The Linux Foundation's policies or the notices set forth on this slide.



CE Workgroup

# Moving Forward

- Expect U of MN to add an experienced kernel developer to their staff, to pre-review commits
- Expect LF TAB to produce a report on best practices for academia to collaborate with OSS communities



# U of MN incident resources

- Good summary of issues:
  - <https://www.zdnet.com/article/university-of-minnesota-security-researchers-apologize-for-deliberately-buggy-linux-patches/>
  - <https://lwn.net/Articles/853717/>
  - <https://lwn.net/Articles/854401/>
  - <https://lwn.net/Articles/854775/>
- TAB produced a very thorough report of the status:
  - <https://lore.kernel.org/lkml/202105051005.49BFABCE@keescook/>
  - UofM and LF TAB are moving forward

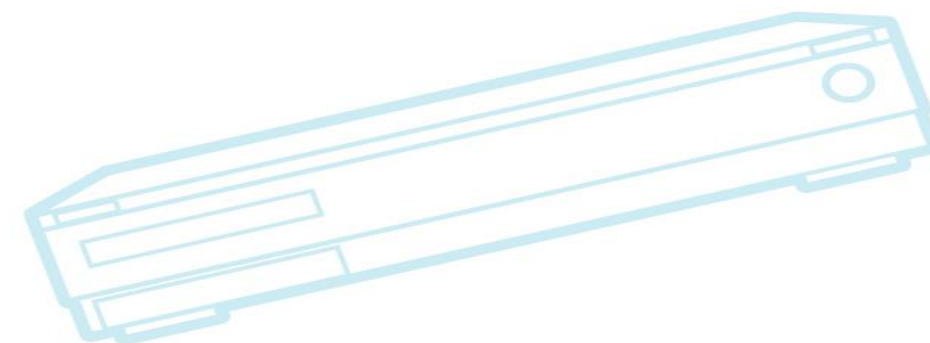
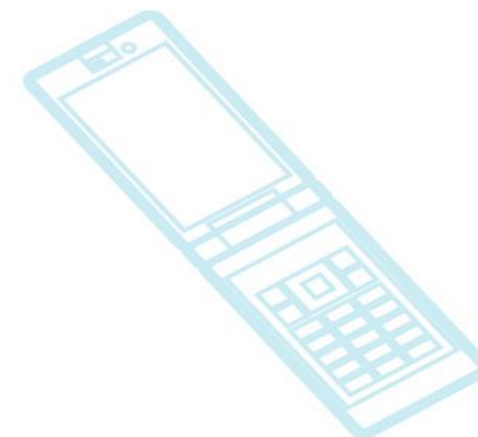
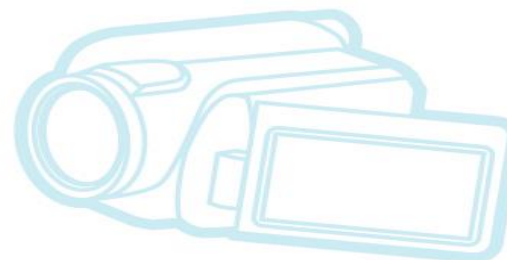




CE Workgroup

# Outline

Linux Kernel  
**Technology Areas**  
Conferences  
Industry News  
Resources





CE Workgroup

# Technology Areas

- Audio
- Core Kernel
- Development
- Filesystems
- Graphics
- Networking
- Security
- Testing
- Toolchains



# Audio

- PipeWire daemon project intends to replace PulseAudio to manage desktop Linux audio mixing and routing
  - Uses time-based audio scheduling (rather than relying on sound card interrupts)
  - Makes extensive use of Linux-only APIs
    - e.g. memfd, eventfd, timefd, signalfd, epoll and dma-bug
  - Supports raw ALSA applications via a plugin
  - Fedora 34 will replace PulseAudio with PipeWire
    - Other distros have the package, but have not made it the default choice
    - We'll see how it goes



# Core Kernel

- /dev/kmem finally removed
  - Was the predecessor to /proc and /sys
  - Used by original 'ps' utility (and others) to read kernel internals
  - Was a massive security problem
  - Has been disabled by most distributions for years
  - Has finally been removed completely
  - See <https://lwn.net/Articles/851531/>
  - On a personal note:
    - My answer on stack overflow about sysfs, and mentioning /dev/kmem is the single largest contributor to my reputation score there
      - /dev/kmem – Rest In Peace!





CE Workgroup

# Development

- New tools are being used for upstream kernel work (especially 'b4'):
  - e-mail workflow of Linux kernel often has issues
    - Some e-mail clients and servers mangle the data
  - b4 supports doing patch management outside of e-mail
    - Acquire a patch series from [lore.kernel.org](http://lore.kernel.org), and use git to apply it
  - See “Kernel Email Tools” talk by Frank Rowands at ELCE 2020
    - [https://elinux.org/images/9/93/Kernel\\_email\\_tools\\_elce\\_2020.pdf](https://elinux.org/images/9/93/Kernel_email_tools_elce_2020.pdf)



CE Workgroup

# Filesystems

- F2FS gets more compression options
  - LZ4 “high compression” mode is supported (v5.12)
  - `compress_mode=` option for whether kernel or user-space controls compression
    - New `ioctl` to give user-space control over which files are stored in compressed form
- `io_uring` continues to mature
  - As a reminder: see <https://lwn.net/Articles/810414/>
  - Supports `tee()` system call (v5.8)
  - Asynchronous buffered reads without using kernel threads (v5.9)
  - Integrated with memory control groups for better accounting and regulation (v5.12)



# Filesystems (cont.)

- Ext4 fast commits (v5.10)
  - Introduces a 2nd “fast commit” journal
    - Holds changes since the “standard commits” journal was updated
  - Fast commit journal has file-level data
    - Omits block data and data that can be recreated from other sources
      - Reduces the amount of data that needs to be saved
  - Fast commits cannot be performed for all ops
    - Falls back to a “standard commit” in this case
  - During recovery, standard commits are replayed by journaling layer, then fast commits are replayed by the filesystem
  - Big performance improvement! (20%-200%)
  - See <https://lwn.net/Articles/842385/>



CE Workgroup

# Graphics

- Generic USB Display driver mainlined in v5.13
  - Allows to push graphics and video over USB
  - Can be used to turn a raspberry PI zero into a USB-to-HDMI adapter
    - See <https://github.com/notro/gud/wiki>
  - See [https://www.phoronix.com/scan.php?page=news\\_item&px=Generic-USB-Display-GUD-5.13](https://www.phoronix.com/scan.php?page=news_item&px=Generic-USB-Display-GUD-5.13)

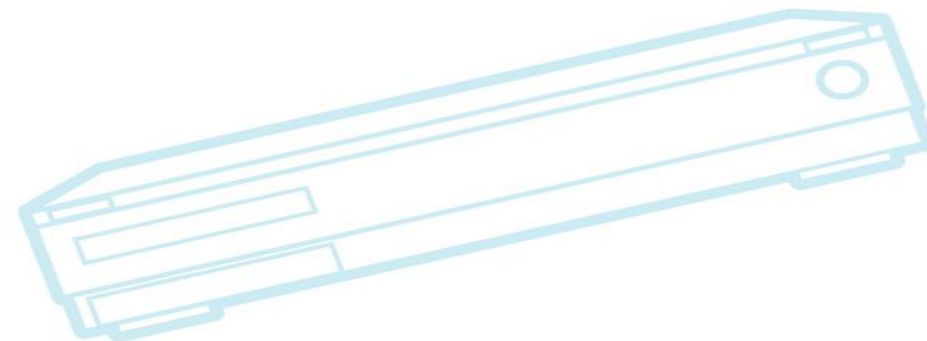
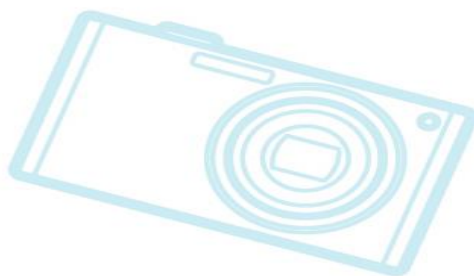
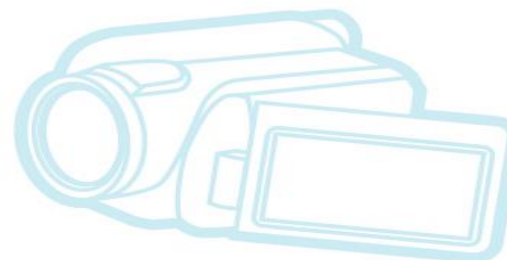
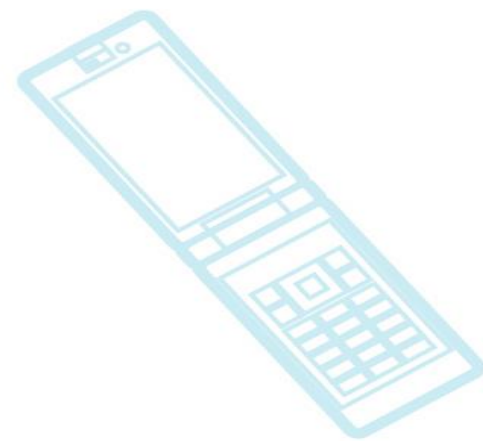




CE Workgroup

# Languages

- Rust
- Python





CE Workgroup

# Rust in the Linux kernel

- People are working to support the Rust language for kernel driver development
  - Have been talking about it for a while
  - Posted RFC to kernel mailing list in April
- Big advantage (claimed) is memory safety features of Rust built in to the language
- Reaction of kernel developers is mixed:
  - Don't want to require kernel developers to know more languages
  - Not sure benefit is worth the cost
  - Most maintainers seem to have a “wait and see” attitude



# Rust in the Linux kernel (cont.)

- Status:
  - Right now, it's only targeted at device drivers
  - Will take years to develop supporting infrastructure and libraries to handle mismatch between C and Rust code and memory models
  - But, Linus Torvalds wrote: "on the whole, I don't hate it"
  - Developers (including Linus) really want to see examples where the Rust code provides benefits, and is not harder to maintain
    - In a real driver - not just a demo or sample
- See <https://lwn.net/Articles/853423/>



CE Workgroup

# Python

- Pyodide project allows running python applications in a web browser
  - Requires integration of JavaScript (which is well-supported by browsers) and Python
  - Project started in 2019 to create the full Python data science stack in the browser
    - Compiles CPython and various data science libraries (e.g. NumPy) to WebAssembly
  - See <https://lwn.net/Articles/855875/>
  - I was sad when Netscape adopted Javascript as the core web language, instead of Python (way back in 1995)
    - Maybe Python can make a comeback in the browser





# Networking

- Wireless WAN (WWAN) framework added to kernel (v5.13)
  - Also known as “Mobile Broadband”
  - Usually provided by cellular networks such as 2G, 3G, 4G LTE or 5G and cellular modems
  - First user is a Qualcomm wwan modem driver
  - Adds the concept of a WWAN port, which is a logical pipe to a modem control protocol
  - See code in drivers/net/wwan
    - Unfortunately, there doesn't seem to be any docs under Documentation
  - See [https://www.phoronix.com/scan.php?page=news\\_item&px=Linux-WWAN-Subsystem-v9](https://www.phoronix.com/scan.php?page=news_item&px=Linux-WWAN-Subsystem-v9)



CE Workgroup

# Networking

- WiMAX code was removed in v5.13
  - Was to support 802.16 / WiMAX, which never took off
    - Protocol was championed by Intel in 2008
    - Intended to support rural areas
  - Also removed single driver for this protocol
  - Interesting to see items removed from kernel



# Real-Time

- Can disable process migration between CPUs (v5.11)
  - Good for keeping an real-time process pinned on a particular CPU
  - Turns out to be hard (who knew?)
  - See <https://lwn.net/Articles/836503/>
- PREEMPT\_DYNAMIC allows selecting preemption mode at boot time or run time (v5.12)
  - Can be “none”, “voluntary”, or “full”
  - There’s an option under debugfs for controlling the mode at run time



CE Workgroup

# PREEMPT\_RT status

- Software-interrupt processing code from Preempt RT was mainlined in v5.13
  - Software interrupts can now be preempted by higher-priority processes (like other parts of the system)
- What's left in Preempt RT to mainline:
  - About 10,000 lines of code (in 199 patches)
    - Fixes to locking code
      - Add locking rtmutex and finishes sleeping spinlocks
    - Some big changes to printk
    - Changes to slub memory allocator
  - People are anxious for Linux RT without having to apply a patch





# PREEMPT\_RT status (cont.)

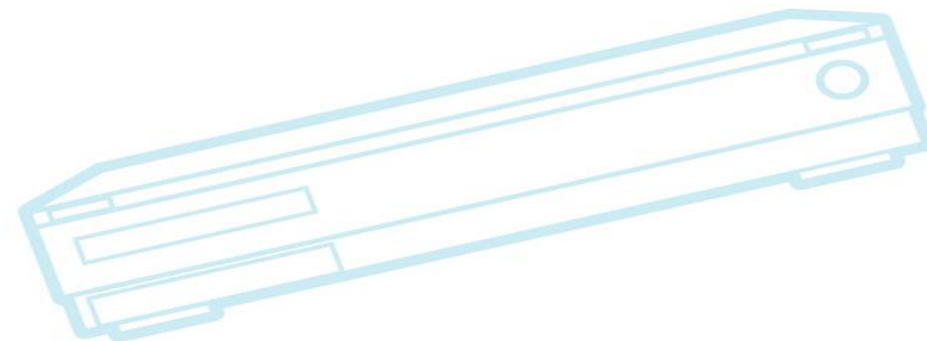
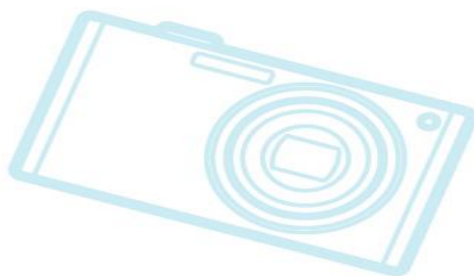
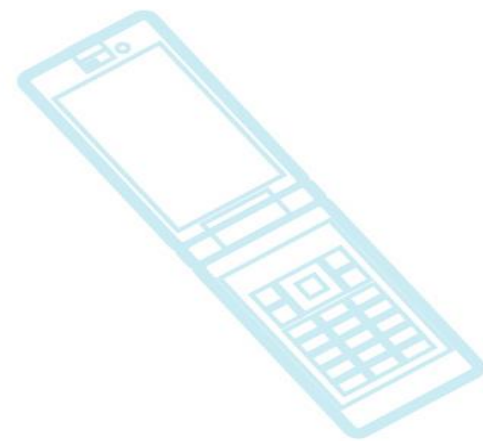
- Always need more funding
  - Infrastructure work never gets the same resources as the latest trendy item
- After PREEMPT\_RT patch mainlining is done, still need to:
  - Maintain RT features so that new kernel code doesn't break realtime
  - Enable RT support for kernel features that are currently disabled under RT
- See Interview with Thomas Gleixner
  - <https://www.linux.com/news/in-the-trenches-with-thomas-gleixner-real-time-linux-kernel-patch-set/>



CE Workgroup

# Security

- Inline encryption for filesystems
- Control-flow integrity
- Landlock security module





CE Workgroup

# Inline encryption for filesystems

- Allows kernel to offload encryption and decryption to the storage device
- Normally, encrypted filesystem has lots of overhead
  - On main CPU or in crypto hardware and busses as data is copied multiple times
- This is faster
  - Kernel manages setup, then storage device handles cryptography
- Kernel can also validate that encryption worked as expected
- See <https://lwn.net/Articles/797309/>
  - Added in v5.8



# Control-Flow Integrity (CFI)

- Mainlined in v5.13 kernel
- CFI checks that indirect call goes to function with same signature as expected
  - At the start of the function
- Detects if malicious code has changed the destination site for an indirect call
  - There are literally thousands of indirect calls in the kernel
- Feature merged in v5.13 is referred to as “forward-edge CFI”
  - There are separate mechanisms for guarding returns (“backward-edge CFI”)





CE Workgroup

# Control-Flow Integrity (cont.)

- Requires Clang and Link Time Optimization
  - All indirect call targets are analyzed at compile time and put into jump tables
    - Loadable modules are handled separately
- Check is made at execution time that indirect call is to a valid target
  - Incurs less than 1% overhead (claimed)
- ARM64 supported now
  - x86 support is in the works
- See <https://lwn.net/Articles/856514/> and <https://lwn.net/Articles/810077/>



CE Workgroup

# Landlock security module

- Landlock security module (new in v5.13)
  - Allows a process to be converted to a secure mode where the kernel can do additional validation of file system operations
  - Allows attaching a ruleset to a process, that allows the system to manage access control to files
  - Allows regular users to create secure execution environments
    - Not just sysadmins and root
  - Accepting a new security module is a big deal
    - There are not many Linux security modules in the kernel
    - This one took over 5 years and 74 revisions to get accepted
  - See <https://lwn.net/Articles/703876/>



CE Workgroup

# Landlock sample

- See demonstration program in kernel source tree at `samples/landlock/sandboxer.c`
  - Set `LANDLOCK_ALLOWED` environment variable to a list of directories
  - Start a program with the 'sandboxer' app
  - Program can only access files under the specified directories



# Testing

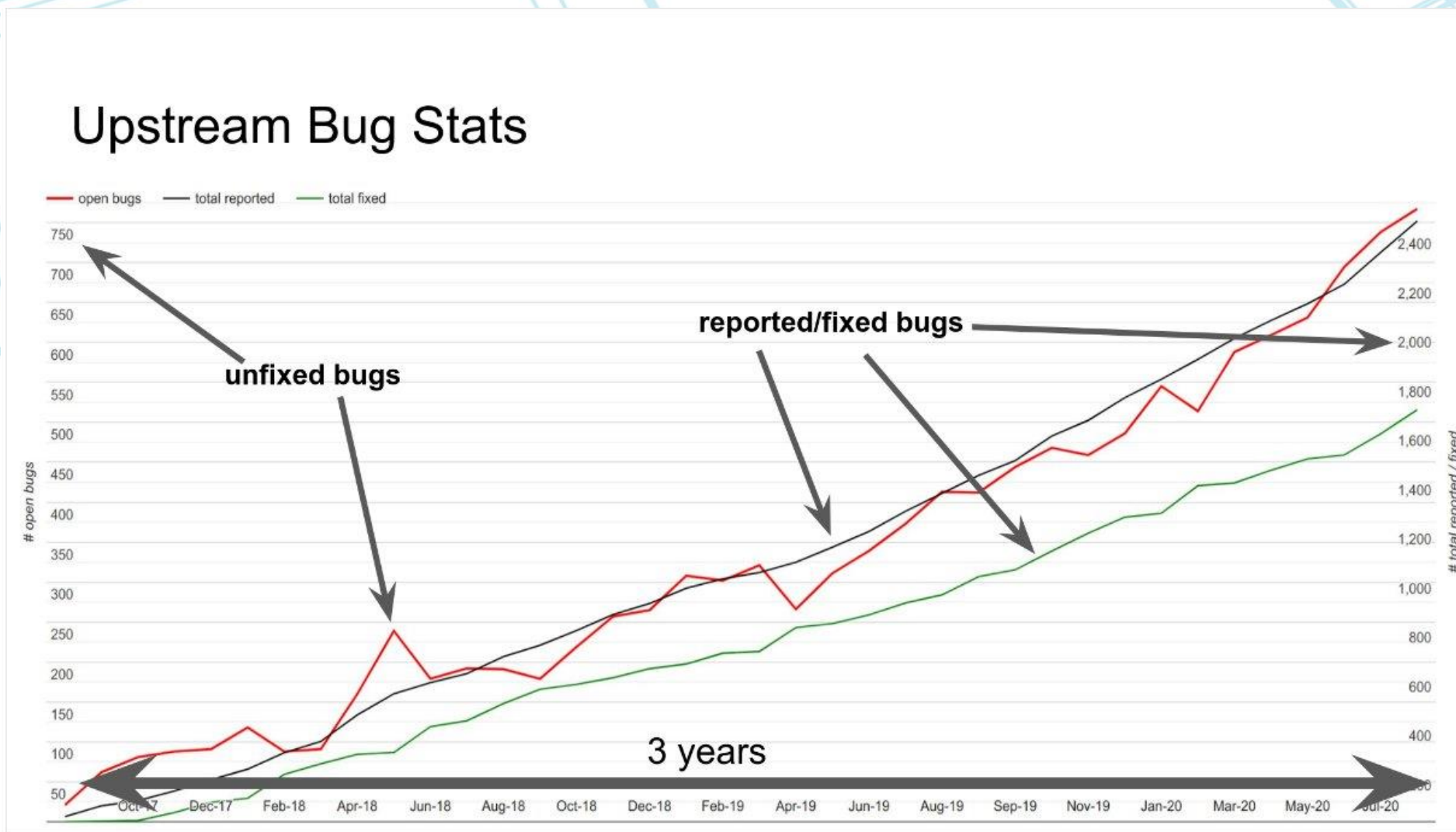
- Syzbot fuzzer continues to find an alarming number of bugs
  - Number of unfixed bugs climbs every release
    - Over 2400 open bug reports (just from syzbot)
    - Some were reported years ago
    - Need to be fixing more bugs than we find
  - See [https://github.com/ossf/wg-securing-critical-projects/blob/main/presentations/The\\_state\\_of\\_the\\_Linux\\_kernel\\_security.pdf](https://github.com/ossf/wg-securing-critical-projects/blob/main/presentations/The_state_of_the_Linux_kernel_security.pdf)





CE Workgroup

# Syzbot Upstream Bug Chart





# Compass CI – new automated test framework

- Compass CI = Huawei's version of zero-day
  - Is used by Huawei to test OpenEuler – which is their Linux distro
  - Create by Fengguang Wu, who created 0day and the LKP kernel test robot for Intel
  - Has lots of features to make it suitable for developer access to hardware and software under test
    - User can select architecture, hardware, distribution, version of distribution, and set of tests
    - Can also get interactive access to hardware
      - Single command to reserve hardware and get ssh access
    - Vision to standardize on sharing various test artifact and interface to create a testing ecosystem
- See <https://connect.linaro.org/resources/lvc21/lvc21-202/>



CE Workgroup

# Toolchains

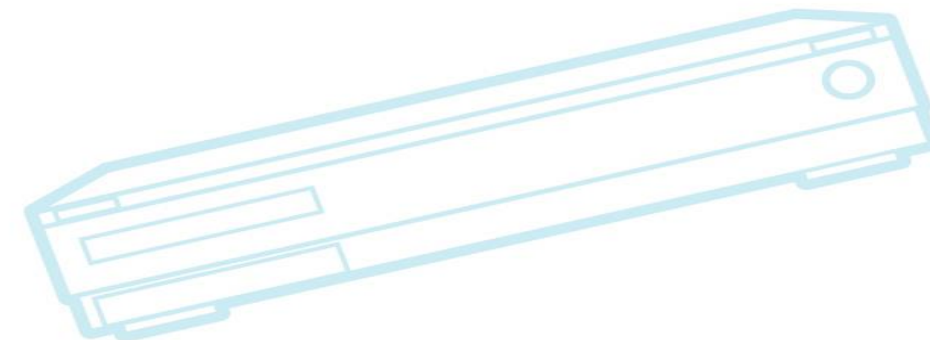
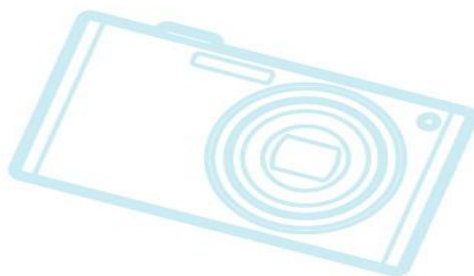
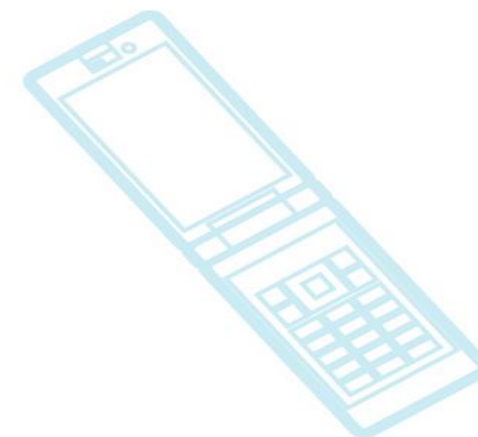
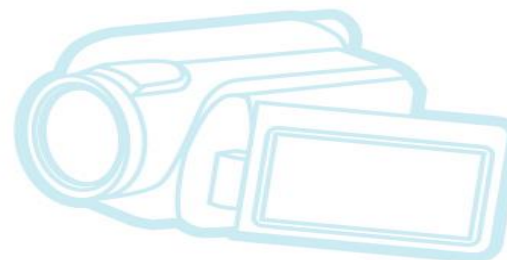
- Now possible to build full embedded Linux system with Clang (LLVM v10)
  - Can build Linux kernel with Clang (for a while now)
  - Harder to build the distribution
  - There's a 'meta-clang' layer available to do builds in Yocto Project
    - Some individual packages will have problems
      - Still need gcc for glibc (use musl C library instead)
  - Debian clang project (<https://clang.Debian.net>)
    - 96% of packages build
  - See talk by Khm Raj at ELCE 2020
    - [https://elinux.org/images/3/3a/ELC\\_2020\\_clang.pdf](https://elinux.org/images/3/3a/ELC_2020_clang.pdf)



CE Workgroup

# Toolchains (cont.)

- LLVM 12.0 released (April 14, 2021)
  - Lower clangd memory usage
  - Improvements to Clang's diagnostics
- GCC 11 released (April 27, 2021)
  - Link-time optimization improvements



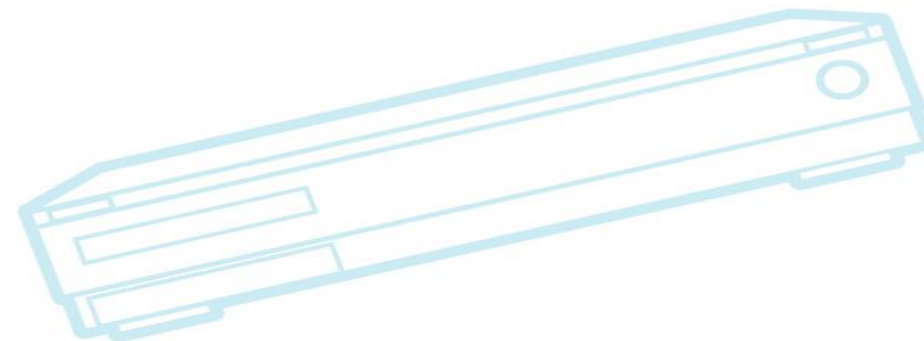
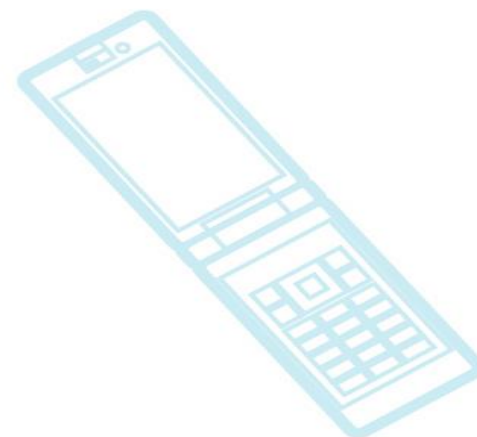
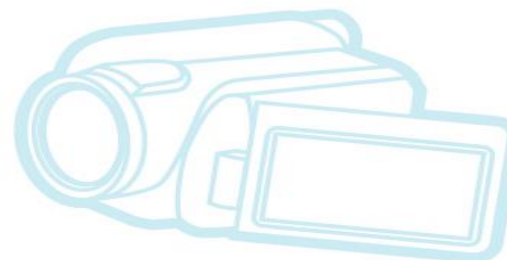




CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
**Conferences**  
Industry News  
Resources

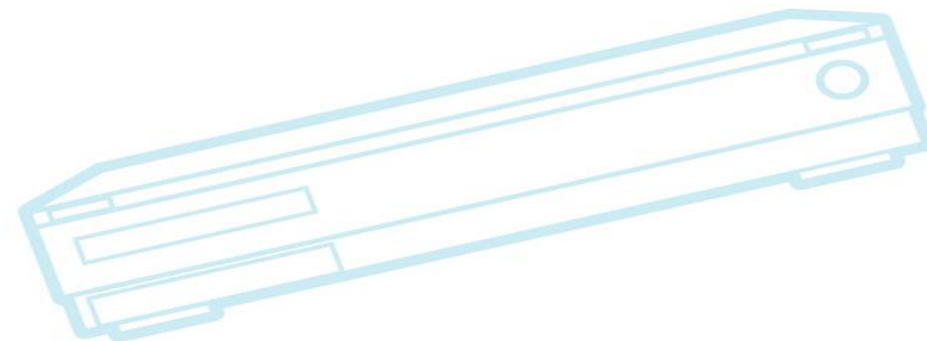
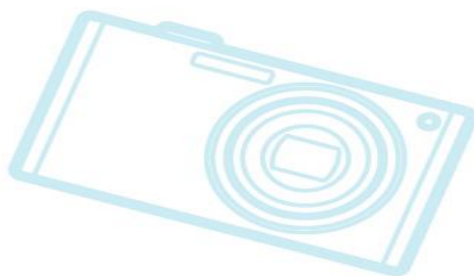
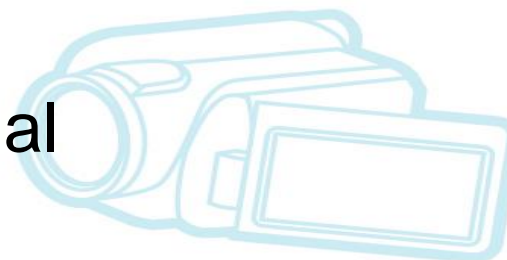
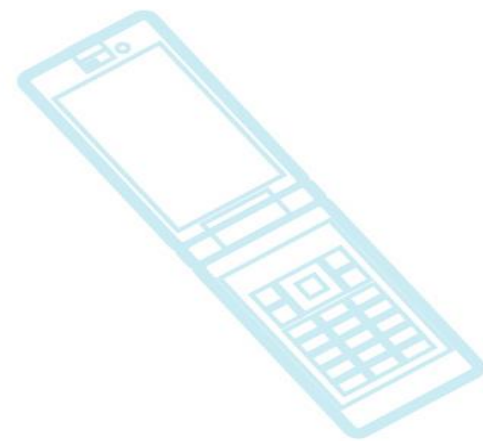




CE Workgroup

# Conferences – 2020

- ELC Europe 2020
  - Virtual in October
- Open Source Summit Japan
  - Changed to December, and went virtual





CE Workgroup

# Conferences - 2021

- Embedded Linux Conference North America
  - ~~August in Vancouver, Canada~~
- Embedded Linux Conference 2021
  - ~~September 28-October 1, Dublin, Ireland (??)~~
  - September 27-30, Seattle, Washington, USA
    - Was “ELC Europe”, but that doesn’t make sense any more
  - Will be a hybrid in-person and virtual event
  - CFP closes on June 13
    - Please propose a session! You don’t have to travel to attend.
  - Timezone will be Pacific time
    - Will be late for Europe and Africa, and inconvenient for Asia



# COVID-19 issues?

- When will conferences no longer be cancelled?
  - ELC 2021 (in late September) is first LF in-person event on schedule
  - Situation is changing as people get vaccinated
    - However, there are a lot of positive signs
  - Things are changing very rapidly in the US
    - Many venues have no masks, and less social distancing
      - Outdoors in my area it is rare to see a mask
- Events will continue with hybrid style for many months, even after in-person events start





CE Workgroup

# Hybrid Events

- Events will continue supporting virtual access
  - Improvements for virtual attendees
    - e.g. live Q&A during sessions
  - Can also speak virtually, even if event is live
- On-site changes for in-person health consciousness
  - Reduced attendance
  - Social distancing
  - Masks, extra cleaning
  - Changes to format of some event activities



CE Workgroup

# Conferences – misc.

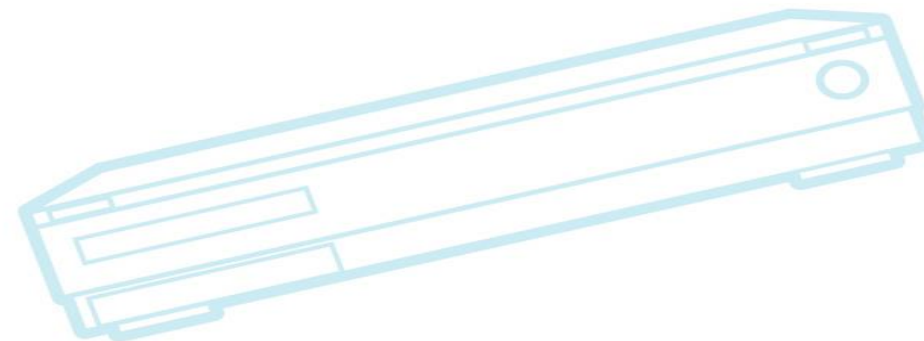
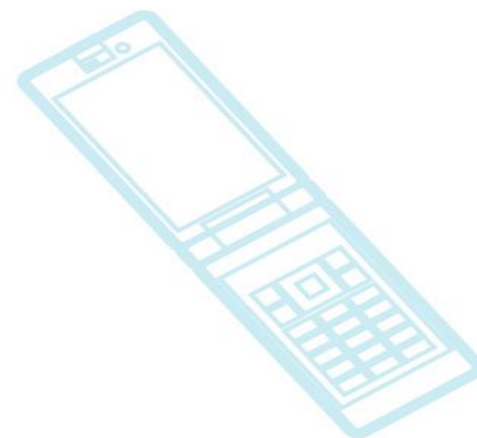
- LF continuing to push for inclusion:
  - Have programs to encourage outreach
  - Diversity training, etc.
- Tux Turns 30 this year! (September, 2021)
  - Would be nice to have some big parties, but we'll see what happens



CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Conferences  
**Industry News**  
Resources

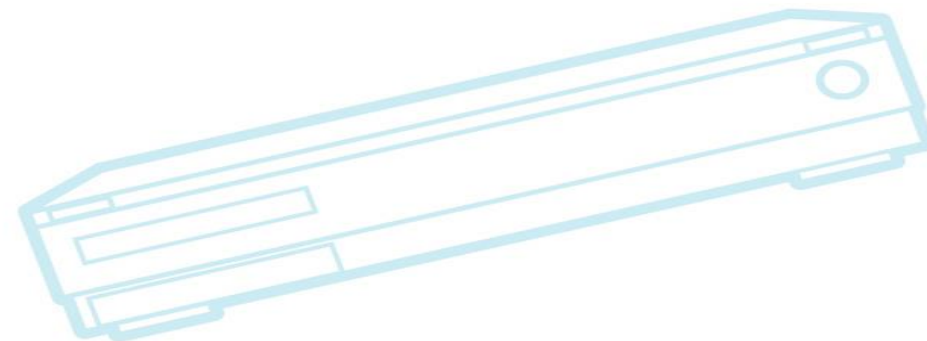
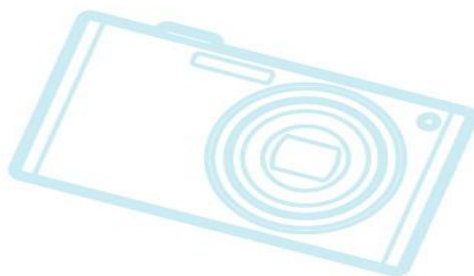
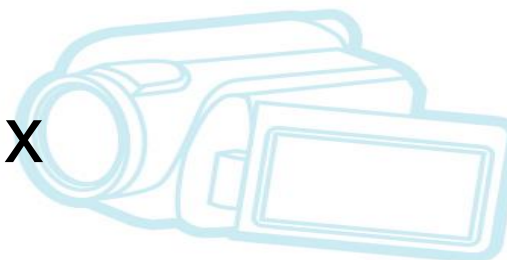
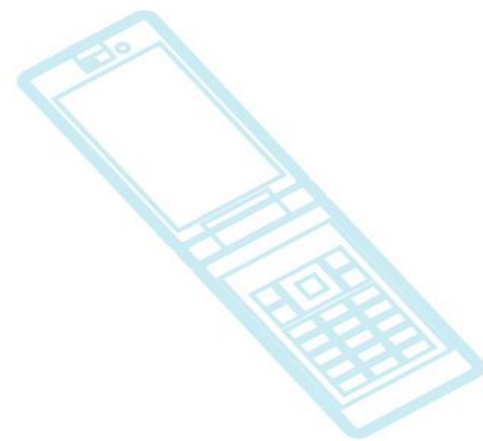




CE Workgroup

# Industry News

- Trade associations
  - Linux Foundation
- Google vs. Oracle
- Interesting cases of embedded Linux







CE Workgroup

# Linux Foundation

- Financials looking very good despite COVID
- More than 1 new member per day
- Training and mentorship has really ramped up:
  - 2M trainings and exams delivered as of Jan 2021
  - New COBOL training
- LFX tools for managing projects
  - Project insights, security, mentorship, crowdfunding, events, training, control center
  - Some tools online now
  - See <https://lfx.dev/>



# Linux Foundation stats

We continue to scale our communities on every level by automating processes, creating innovative developer tools, and focusing on community value.



**11.7M**

Lines of Code  
Added Weekly



**9.4M**

Lines of Code  
Removed Weekly



**208,660**

Contributing  
Developers



**12,435**

Contributing  
Companies



**16,618**

Repositories



**9.1M**

Commits



**960,610**

Pull Requests



**1.3M**

Builds Monitored



**859,150**

Logged Issues



**5.2B**

Container  
Downloads



**1.6M**

Group Chat  
Messages



**3.6M**

Group Email  
Messages



**4,532**

Scanned  
Repositories



**282,342**

Open  
Vulnerabilities



**62,379**

Recommended  
Fixes



**36,163**

Vulnerabilities  
Fixed



**31,496**

CLA  
Contributors



**26,998**

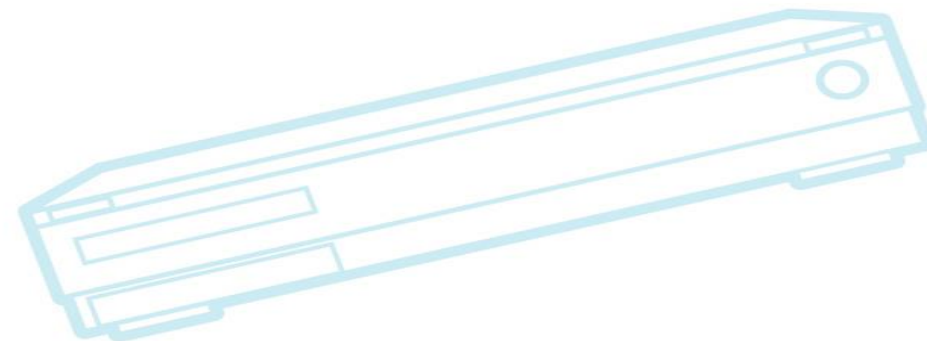
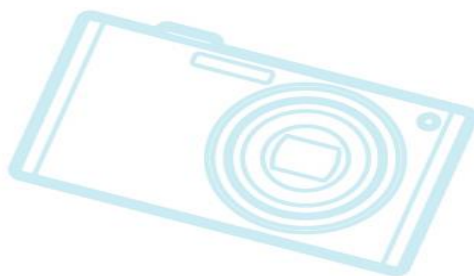
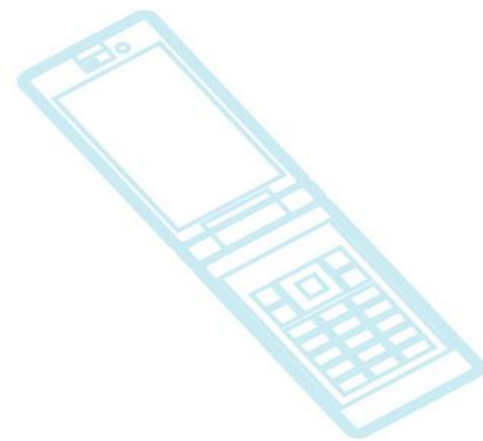
Community  
Meetings



CE Workgroup

# LF Projects

- Some recent new initiatives
  - Linux Foundation Research
  - AgStack Foundation
  - Green Software Foundation







CE Workgroup

# Linux Foundation Research

- New initiative to measure, analyze and describe the impact of open source collaborations
- Will use data from LF projects and tools (e.g. LFX), as well as other sources, to prepare reports
- Already hired Hilary Carter as new LF VP
  - She did this type of work for Blockchain
- Will also establish an LF Research Advisory Board, to influence the research program agenda
  - Rotating committee of community leaders and subject matter experts
- Look for OSS research reports in the future from LF





CE Workgroup

# AgStack Foundation

- A project to create an open repository of tools, models, data and source code for the Agriculture/Farming sector
  - Will use existing technologies and standards like:
    - Standards: AgGateway, UN-FAO, CAFA, USDA, NASA-AR
    - Public Data: Landsat, Sentinel, NOAA, Soilgrids
    - Models: UC-ANR IPM
    - Code: Hyperledger, Kubernetes, Open Horizon, Postgres, Django, etc.
- Goal is to increase productivity and reduce waste
  - 33% of all food produced worldwide is currently wasted
  - Increased collaboration can help interoperation and improve efficiency throughout the agriculture supply chain



CE Workgroup

# Green Software Foundation

- Build a trusted ecosystem of people, standards, tooling and best practices to creating “green software”
  - Reduce power usage, and carbon emissions of software
- Data centers account for 1% of global electricity demand
  - Forecast to rise to 3-8% in the next decade
- Want to limit the carbon footprint of data centers and other computer-related activities through improvement of software



# Google v Oracle

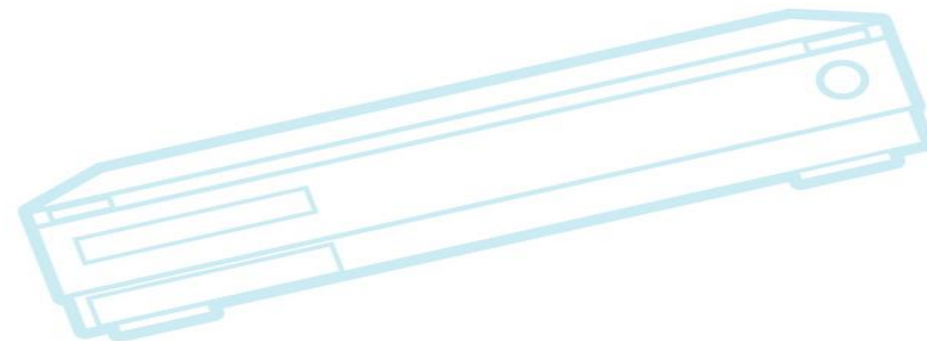
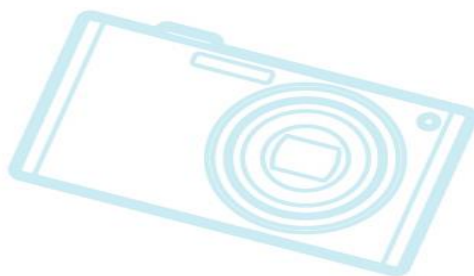
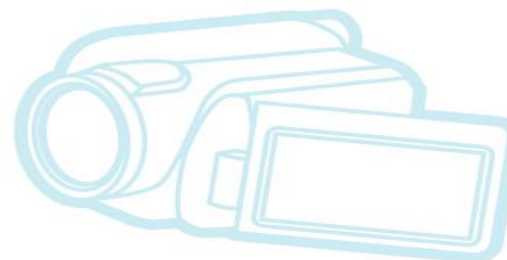
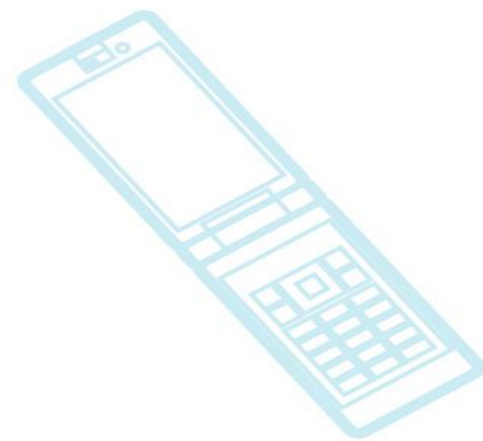
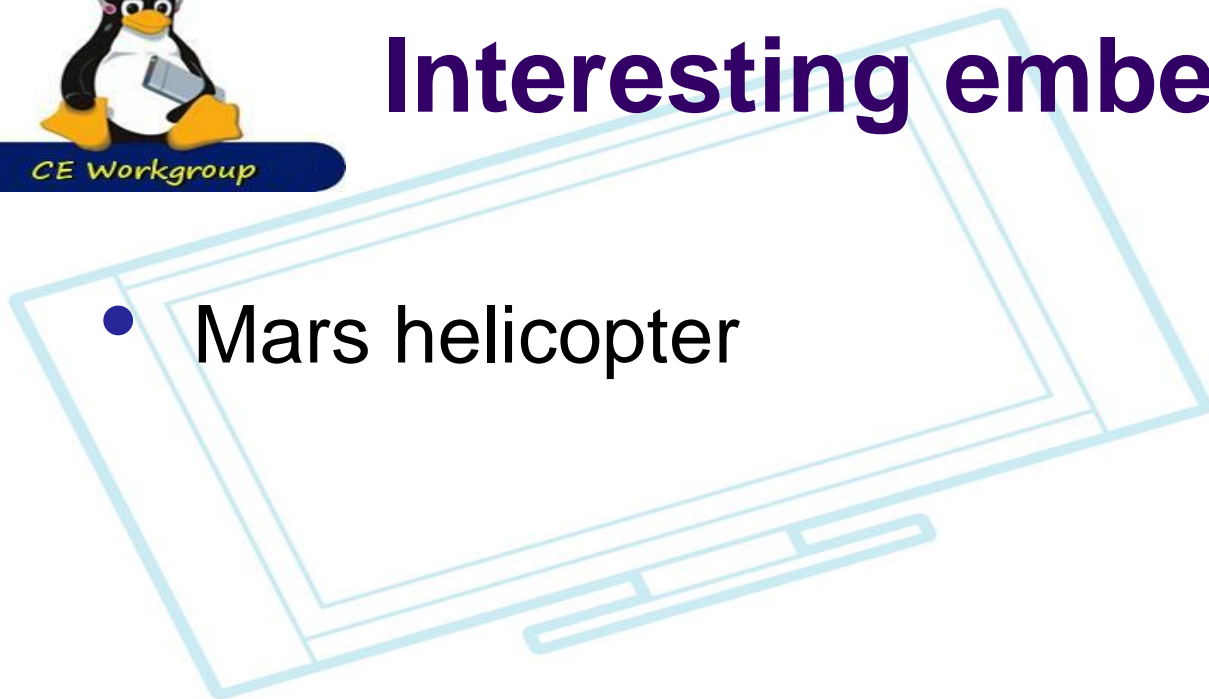
- Supreme Court of the US decided for Google in Google v. Oracle API case (decided April 5, 2021)
  - Case has been going through courts for many years
  - Oracle asserted that Google couldn't use JAVA API in Android
    - That this was a copyright violation
  - Google did not use Oracle's JAVA implementation, but only the API definitions (e.g. for JAVA method call signatures)
  - SC ruled that Google's use was "Fair Use" under US copyright law
    - SC did NOT rule on copyrightability of APIs
  - This is a HUGE deal
    - Most of the industry wanted APIs to be usable in an open way
    - Decision in favor of Oracle would have been very hard for OSS



CE Workgroup

# Interesting embedded Linux

- Mars helicopter







CE Workgroup

# Mars Helicopter - Ingenuity





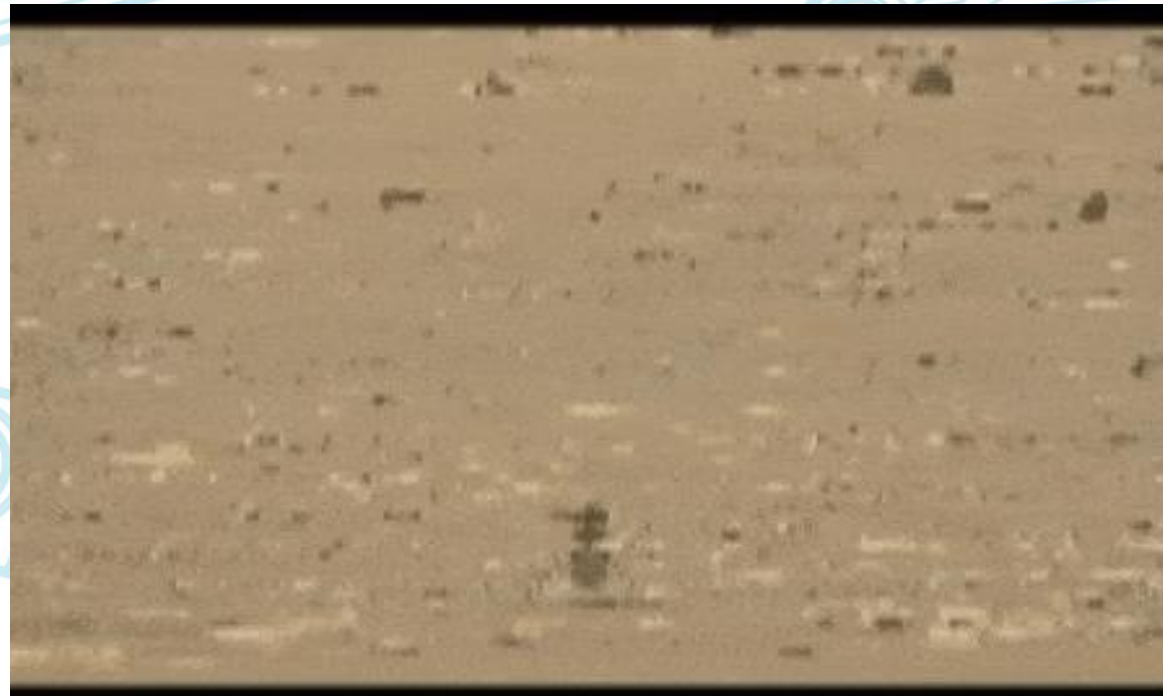
# Mars Helicopter

- Mars Ingenuity Helicopter landed in February on Mars
- Performed tests and demonstrations in April
  - Of use of COTS in space
    - BAE RAD-hardened processor on the rover is about \$250,000 dollars, and runs at 200 MHz (VxWorks)
    - Helicopter has Qualcomm Snapdragon 801 processor, running at 2.6 GHz, with Linux OS
      - Same processor as used in 2014/2015 mobile phones
    - Some components were off-the-shelf:
      - e.g. LIDAR purchased from SparkFun
  - Still high development cost: Total program is about \$80 million dollars, including operations



# Mars helicopter results

- It worked!!!
  - Performed 5 flights – each with increasing difficulty
  - Last flight (flight 5) was a one-way trip to a new location







# Mars Helicopter Hardware

- Hardware:
  - 2x 1.2-meter counter-rotating blades
    - Large, and spinning 5 times faster than Earth drones
      - To deal with thin Martian atmosphere, which is only 1% as dense as on Earth
  - 13 megapixel camera (forward-facing) for 4K video
  - .5 megapixel camera (downward-facing) for terrain mapping and navigation
  - laser altimeter, inertial, tilt and other sensors
  - batteries, solar panels, carbon tube landing legs
- Body of helicopter is insulated and heated so electronics can withstand overnight cold temperatures





# Mars Helicopter Software

- Lots of Open Source
  - Linux operating system
    - Version 3.4
  - Uses “F Prime” OSS flight software, published on github by NASA
    - <https://github.com/nasa/fprime>
    - Can use this flight control software in your own projects
  - List of projects used at:
    - <https://docs.github.com/en/github/setting-up-and-managing-your-github-profile/customizing-your-profile/personalizing-your-profile#list-of-qualifying-repositories-for-mars-2020-helicopter-contributor-badge>
    - If you have a commit in one of those projects, you can get a contributor badge for your github account!



# Mars Helicopter Software (cont.)

- Guidance loops are running at 500 HZ
  - Doing feature tracking from frame to frame at 30 HZ
- Does pre-programmed flight with moderate (not high-level) autonomy
  - Launch is “blind” until 1 meter off ground (using inertial sensors but not camera or altimeter – to avoid dust interference)
  - No GPS on Mars – must use own sensors and estimate position
  - Does not autonomously select targets, or do “smart” navigation
  - Uses OpenCV and pattern matching to establish position and orient helicopter, and to re-find landing site



# Mars Helicopter Software Bugs

- Problems were experienced during first spin-up test, and first attempt of flight 4
  - NASA said these were from a “watchdog timer” expiration
    - There was an unexpected delay transitioning between operating modes
  - NASA modified the instruction sequence to mitigate the issue
    - One plan was to upload new firmware, but they didn’t end up doing that
  - Estimate that each flight has an 15% chance of failure
    - Can still upload new firmware if needed (and it’s already been prepared)
- Other than that, all software has performed as intended





# Mars Helicopter Mission

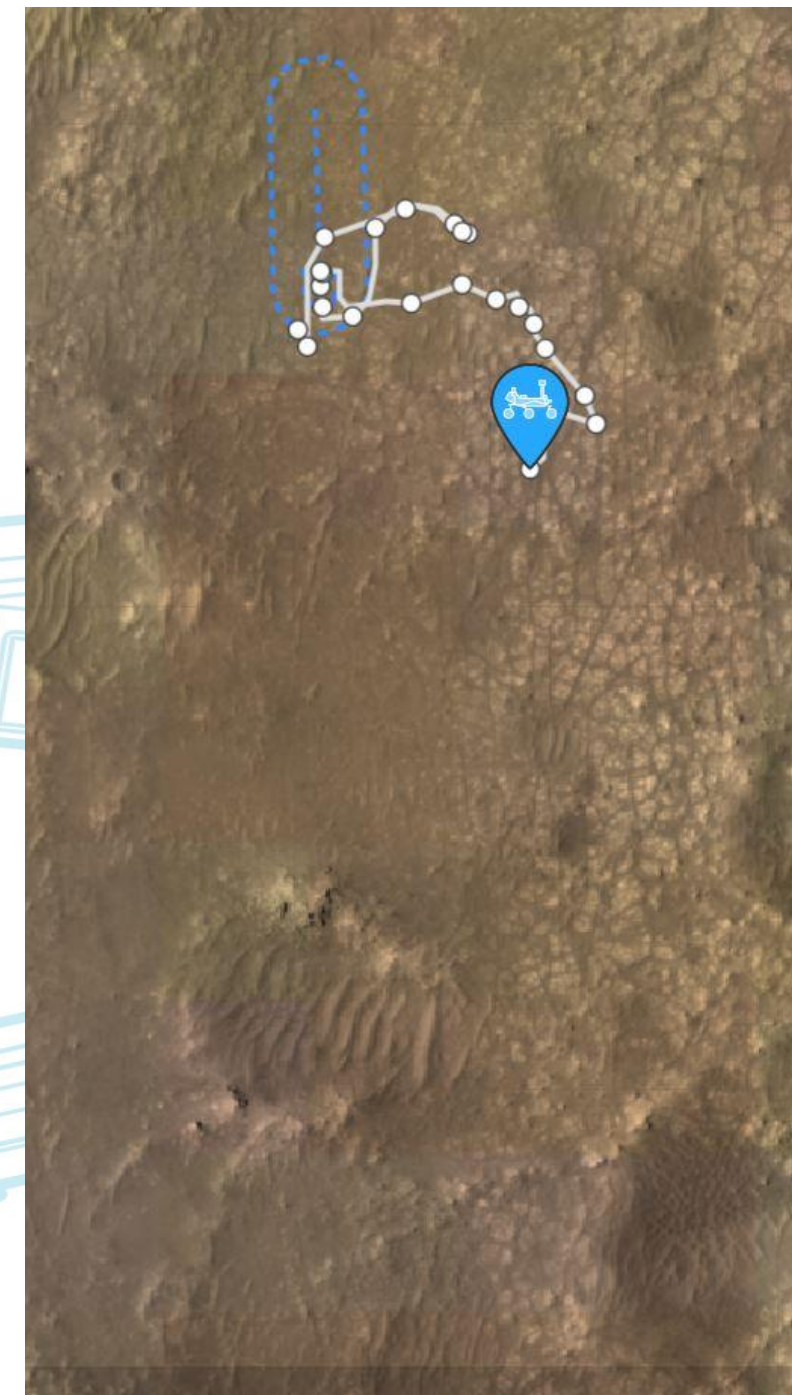
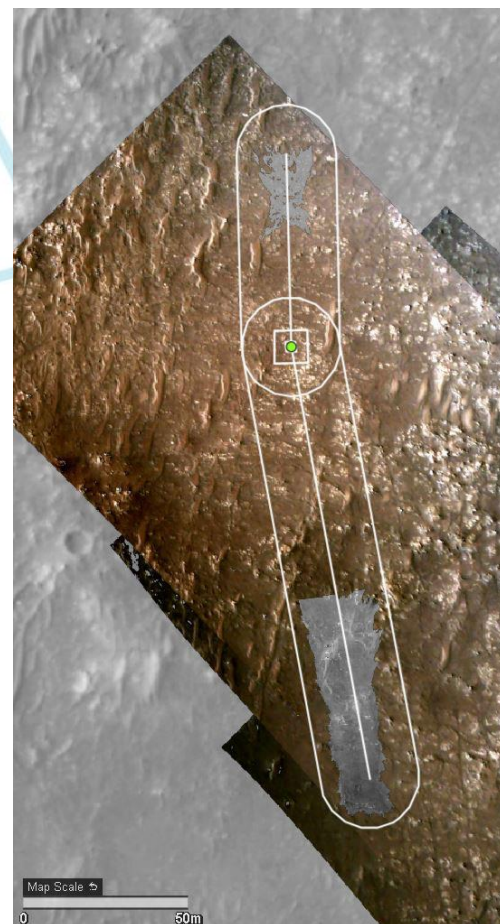
- 30-day window (budget) for first 5 flights (Technology demonstration)
  - Not a lot of Mars science
    - Just a demo that it can be done at all, with COTS and Linux
    - Test hardware/software capabilities in extreme environment of another planet
      - Very thin atmosphere, very cold temperatures, less sun, less gravity, etc.
  - This mission was a complete success
    - They gathered lots of performance data, and audio, pictures, video, etc.





CE Workgroup

# Ingenuity flights on Mars





# Helicopter operation on Mars

- Continuing mission for Ingenuity (“Operations demonstration”)
- NASA plans to continue with flights when time is available
  - Continue one-way flights to new locations (possibly to help scout terrain for the rover)
  - Will try to fly to keep in range of rover
  - Provide ariel photography of features of interest for rover exploration
- Expect to fly about every 2 to 3 weeks
- Will continue that mission until August
- Flight 6 is already planned
  - Plan is to fly south, take some photos of a previously unexplored area, then land in an unscouted location





# Sources for Mars helicopter

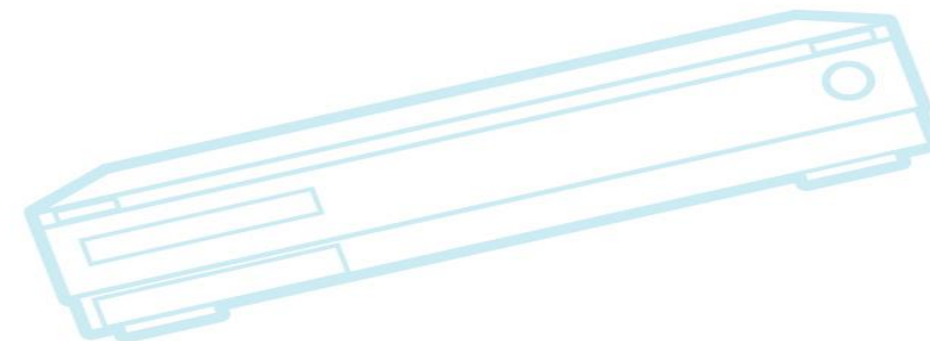
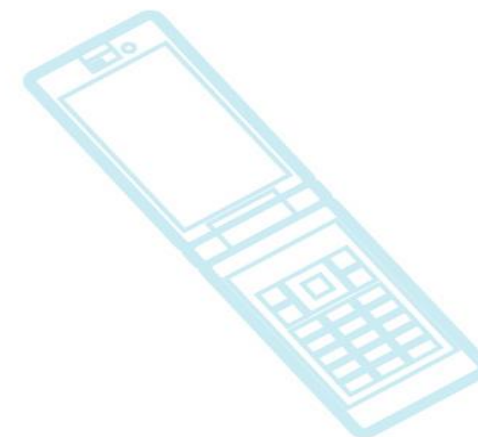
- [https://en.wikipedia.org/wiki/Ingenuity\\_\(helicopter\)](https://en.wikipedia.org/wiki/Ingenuity_(helicopter))
- <https://thenewstack.io/how-the-first-helicopter-on-mars-uses-off-the-shelf-hardware-and-linux/>
- <https://www.pcmag.com/news/4-android-smartphones-with-as-much-power-as-nasas-mars-helicopter>
- [https://www.theregister.com/2021/04/30/ingenuity\\_fourth\\_flight\\_ops/](https://www.theregister.com/2021/04/30/ingenuity_fourth_flight_ops/)
- <https://spectrum.ieee.org/autoton/aerospace/robotic-exploration/nasa-designed-perseverance-helicopter-rover-fly-autonomously-mars>
- <https://mars.nasa.gov/technology/helicopter/status/302/plans-underway-for-ingenuitys-sixth-flight/>



CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Conferences  
Industry News  
**Resources**







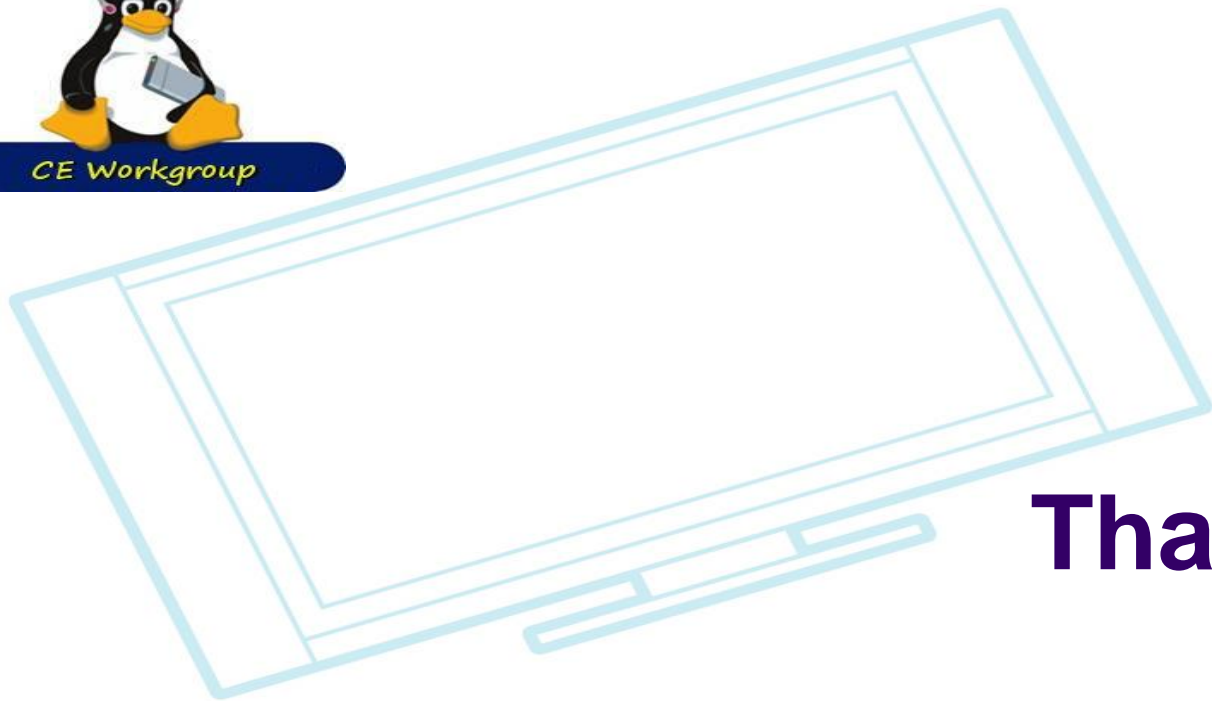
CE Workgroup

# Resources

- LWN.net – <https://lwn.net>
  - If you are not subscribed, please do so
  - Some content is delayed by 2 weeks for non-subscribers (some links in this presentation)
- Phoronix - <https://www.phoronix.com/>
- eLinux wiki – [elinux.org](http://elinux.org)
- Google



CE Workgroup



**Thanks!**

